

UNITED STATES PATENT APPLICATION

for

SYSTEM AND METHOD FOR NETWORK VIRUS PROTECTION

Inventors:

Boris Yanovsky

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90026-1026
(408) 720-8300

File No.: 004619.P001

EXPRESS MAIL CERTIFICATE OF MAILING

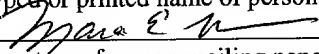
"Express Mail" mailing label number: EL627534036US

Date of Deposit: April 13, 2001

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to BOX PATENT APPLICATION, Assistant Commissioner for Patents, Washington, D. C. 20231

Mara E. Brown

(Typed or printed name of person mailing paper or fee)



(Signature of person mailing paper or fee)

4/13/01

(Date signed)

SYSTEM AND METHOD FOR NETWORK VIRUS PROTECTION

FIELD OF THE INVENTION

[0001] The field of the invention relates to anti-virus protection. More specifically, the invention relates to anti-virus protection of a local area network.

BACKGROUND OF THE INVENTION

[0002] A computer virus is a self-replicating program designed to spread without user intervention or knowledge. Computer viruses are spread by attaching themselves to another program, such as a macro attached to e-mail. A worm is a type of computer virus that can transmit itself to a second computer over a network. The increased access to e-mail at the workplace has allowed viruses and worms to spread at a much faster rate. The number of viruses "in the wild," or present in more than one company or organization, have increased dramatically since widespread Internet access has become available.

[0003] Most companies allow Internet access by creating a local area network (LAN). Access to the LAN by the Internet is protected by a "firewall". Such a network allows programs on one computer to be accessed by all the computers on the LAN.

Unfortunately, this access means that once a virus infects one computer, all the other computers in a LAN may soon be infected as well.

[0004] The standard protection against virus is an anti-virus software application that analyzes software applications and isolates any latent viruses. This anti-virus software has a set of virus characteristics that the software searches for in the computer. Each

time a new virus is created or evolved, a new anti-virus characteristic must be updated to the computer in order for the anti-virus software to detect the virus.

[0005] There are two methods that are used prominently for administering anti-virus software. One method is to install anti-virus software directly into the firewall. The firewall attempts to scan for viruses on the fly while the client computer is receiving the data and then aborts the transfer if a virus is detected. This method has several disadvantages. Having a single point for scanning data creates a bottleneck and slows down the system performance. Additionally, this method only prevents viruses from entering from the Internet and fails to provide protection from viruses distributed locally either through the LAN from one computer to the other or through external media, such as floppy disks.

[0006] The second method is to install an anti-virus client on each individual computer and manage them separately. This protection also has several disadvantages. This method fails to guarantee all the computers on the LAN have the software installed and properly configured, that the virus scanning engine and data files are up-to-date, and that the individual computer user did not disable the anti-virus software.

[0007] What is needed is a method of administering anti-virus applications so that a LAN is protected from both Internet infections and internal infections (from other computers in the LAN) as well. What is further needed is a method of administering anti-virus applications so that scanning engine and data files may be made up-to-date on each computer on the LAN.

SUMMARY OF THE INVENTION

[0008] A system and method for administrating and managing anti-virus protection on a local area network (LAN) is described. In one embodiment, the LAN's anti-virus policy is programmed into an access module. In another embodiment, the access module may be an Internet access module and/or in a firewall. The access module applies the anti-virus policy to client computers on the LAN. In another embodiment, the policy might include the frequency with which the anti-virus software is updated and the number of versions that the software is permitted to be out-of-date. Any client computer not meeting the policy is not permitted to access the Internet. The access module can also update out-of-date client computers to make them compliant with the policy.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which

[0010] Figure 1 illustrates one embodiment of a configuration of a local area network.

[0011] Figure 2 is a flow diagram illustrating one embodiment of a process for updating the anti-virus protection based on time.

[0012] Figure 3 is a flow diagram illustrating one embodiment of a process for updating the anti-virus protection based on software version number.

[0013] Figure 4 is a flow diagram illustrating one embodiment of a process for determining if a host device's anti-virus protection has been disabled.

[0014] Figure 5 is a flow diagram illustrating one embodiment of a process for overriding the tolerances during an emergency virus alert.

[0015] Figure 6 is one embodiment of a computer system.

DETAILED DESCRIPTION

[0016] A system and method for enforcing and maintaining anti-virus protection policies for computers on a local area network (LAN) is disclosed. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one of ordinary skill in the art that these specific details need not be used to practice the present invention. In other circumstances, well-known structures, materials, circuits, processes and interfaces have not been shown or described in detail in order not to unnecessarily obscure the present invention.

[0017] In the following description, numerous details are set forth, such as distances between components, types of molding, etc. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

[0018] Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of

being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0019] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0020] The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or

optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

[0021] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

[0022] A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory ("ROM"); random access memory ("RAM"); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

Overview

[0023] In one embodiment illustrated by the block diagram of Figure 1, the system administrator for the LAN 100 sets the policy for anti-virus protection for the LAN and installs this policy into an Internet access module (IAM) 110. This Internet access module 110 may comprise a live firewall, a proxy server, a router, a modem, a gateway,

or an application server. The IAM 110 then enforces and maintains the anti-virus policy, updating the anti-virus on the host devices 120, such as personal computers, where needed. In one embodiment, the IAM 110 denies access to the Internet 130 for any of host devices 120 that fail to meet the requirements, within some preset tolerances, for the anti-virus policy. The IAM 110 sends updates to the anti-virus protection (AVP) on the host device. In one embodiment, if the AVP is disabled for a host device, the IAM 110 instructs the user to enable the AVP.

[0024] In one embodiment, the system administrator sets the tolerances. These tolerances may include the oldest permissible version of the software, longest time without an update, and, in specific cases, necessary virus search identifiers. In one embodiment, the tolerances under the anti-virus protection policy can differ between two host devices on the same local area network. In one embodiment, these tolerances are checked when the host device tries to access the Internet. In an alternative embodiment, the IAM can use an out-of-band protocol, based, for example, on a user datagram protocol (UDP), to test what version or when the last update was installed upon a host device. In one embodiment, to prevent corruption by a viral agent, communications over the out-of-band protocol are encrypted.

[0025] Figure 2 is a flow diagram illustrating one embodiment of a method for updating the anti-virus protection based on time. The process is performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as is run on a general purpose computer system or a dedicated machine, etc.), or a combination of both.

[0026] Referring to Figure 2, the IAM sends an update query to the host device (HD) as to the status of the host device's anti-virus protection (AVP) (processing block 200). Such a query may be sent at any time. In one embodiment, the update query may be sent when a personal computer is turned on or joins a network. The host device responds to the query with the status of the AVP (processing block 210). In one embodiment, the response includes a timestamp (TS). The IAM checks the timestamp against the last available (processing block 220). If the timestamp is not less than the available time minus the preset time tolerance (TT), then the IAM grants the host device Internet access (processing block 230). If the timestamp is less than the current time minus the preset time tolerance, then the IAM denies the host device Internet access (processing block 240). In one embodiment, the IAM can then send to the host device the software components needed to update the host device's AVP (processing block 250) and sets the timestamp to the current time (processing block 260). Thereafter, the host device is granted Internet access by the IAM (processing block 230).

[0027] The IAM may also send commands. In one embodiment, these commands include, for example, a command to request status of the anti-virus protection of the at least one host device, a command to have the at least one host to update the anti-virus protection, a command to uninstall the anti-virus protection, and a command to check a specific file or directory for a virus, for example.

[0028] Figure 3 is a flow diagram illustrating one embodiment of a method for updating the anti-virus protection based on software version number. The process is performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.),

software (such as is run on a general purpose computer system or a dedicated machine, etc.), or a combination of both.

[0029] Referring to Figure 3, the IAM again sends an update query to the host device (HD) as to the status of the host device's anti-virus protection (AVP) (processing block 300). The host device responds to the query with the status of the AVP (processing block 310). In one embodiment, the response includes a version number (V0) for the AVP software. The IAM checks the version number against the current version number (CV0) (processing block 320). If the version number is not less than the current version number minus the version tolerance (VT), then the IAM grants the host device Internet access (processing block 330). If the version number is less than the current version number minus the preset version tolerance, then the IAM denies the host device Internet access (processing block 340). In one embodiment, the IAM can then send to the host device the software components needed to update the host device's AVP (processing block 350) and sets the version to the current version number (processing block 360). Thereafter, the host device is granted Internet access by the IAM (processing block 330).

[0030] Figure 4 is a flow diagram illustrating one embodiment of a method for determining if a host device's anti-virus protection has been disabled. The process is performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as is run on a general purpose computer system or a dedicated machine, etc.), or a combination of both.

[0031] Referring to Figure 4, processing logic in the IAM sends a status query to the host

device (HD) as to the host device's anti-virus protection (AVP) (processing block 400).

The host device responds to the query with the status of the AVP (processing block 410). Processing logic determines whether the AVP is fully enabled. If the AVP is fully enabled, then processing logic in the IAM grants the host device Internet access (processing block 430). If the AVP is not fully enabled, then processing logic in the IAM denies the host device Internet access (processing block 440) and sends a message to the user of the host device to inform the user that the AVP on that host device is disabled (processing block 450). Processing logic in the IAM then sends to the host device the software components needed to enable the host device's AVP or provides the user with instructions as to enabling the AVP of the host device (processing block 460). After the AVP is enabled, processing logic in the IAM grants the host device Internet access (processing block 430).

[0032] In one embodiment, in case of an emergency situation, where the AVP needs to be updated immediately, the tolerances set by the administrator can be overridden. As illustrated in Figure 5, in one embodiment, the system administrator sends a virus warning to the IAM (processing block 500). Processing logic in the IAM then suspends all Internet access (processing block 510) and proceeds to update all host devices on the LAN (processing block 520). After the host devices have been updated, processing logic in the IAM enables Internet access to resume (processing block 530). In one embodiment, Internet access for a device may resume once that device has been updated.

[0033] Figure 6 is a block diagram of an exemplary computer system that may perform one or more of the operations described herein. Referring to Figure 6, computer system 600 may comprise an exemplary client 650 or server 600 computer system. Computer system 600 comprises a communication mechanism or bus 611 for communicating information, and a processor 612 coupled with bus 611 for processing information. Processor 612 includes a microprocessor, but is not limited to a microprocessor, such as, for example, Pentium™, PowerPC™, Alpha™, etc.

[0034] System 600 further comprises a random access memory (RAM), or other dynamic storage device 604 (referred to as main memory) coupled to bus 611 for storing information and instructions to be executed by processor 612. Main memory 604 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 612.

[0035] Computer system 600 also comprises a read only memory (ROM) and/or other static storage device 606 coupled to bus 611 for storing static information and instructions for processor 612, and a data storage device 607, such as a magnetic disk or optical disk and its corresponding disk drive. Data storage device 607 is coupled to bus 611 for storing information and instructions.

[0036] Computer system 600 may further be coupled to a display device 621, such as a cathode ray tube (CRT) or liquid crystal display (LCD), coupled to bus 611 for displaying information to a computer user. An alphanumeric input device 622, including alphanumeric and other keys, may also be coupled to bus 611 for communicating information and command selections to processor 612. An additional

user input device is cursor control 623, such as a mouse, trackball, trackpad, stylus, or cursor direction keys, coupled to bus 611 for communicating direction information and command selections to processor 612, and for controlling cursor movement on display 621.

[0037] Another device that may be coupled to bus 611 is hard copy device 624, which may be used for printing instructions, data, or other information on a medium such as paper, film, or similar types of media. Furthermore, a sound recording and playback device, such as a speaker and/or microphone may optionally be coupled to bus 611 for audio interfacing with computer system 600. Another device that may be coupled to bus 611 is a wired/wireless communication capability 625 to communication to a phone or handheld palm device.

[0038] Note that any or all of the components of system 600 and associated hardware may be used in the present invention. However, it can be appreciated that other configurations of the computer system may include some or all of the devices.

[0039] Thus, a software application system is described which enforces and maintains local area network anti-virus policies. Although the present invention is described herein with reference to a specific preferred embodiment, many modifications and variations therein will readily occur to those with ordinary skill in the art. Accordingly, all such variations and modifications are included within the intended scope of the present invention as defined by the following claims.